

Laboratoire de logiques floue et applications (LFA)

EVO-TORCS

Temps à disposition : 8 périodes

Objectifs pédagogiques

- Comprendre et utiliser un algorithme génétique simple afin de résoudre un problème d'optimisation.
- Implémenter une méthode d'apprentissage supervisé appliquée à la logique floue.

Cahier des charges

Le but de ce laboratoire est d'implémenter une méthode d'apprentissage supervisé pour système flou, à l'aide d'un algorithme génétique et ainsi réaliser l'apprentissage automatique de 2 systèmes flous contrôlant la direction et l'accélération/freinage d'une voiture de course.

Pour ce faire, vous aurez à disposition les packages jFuzzyLogic, JGAP et un dataset (jeu de données).

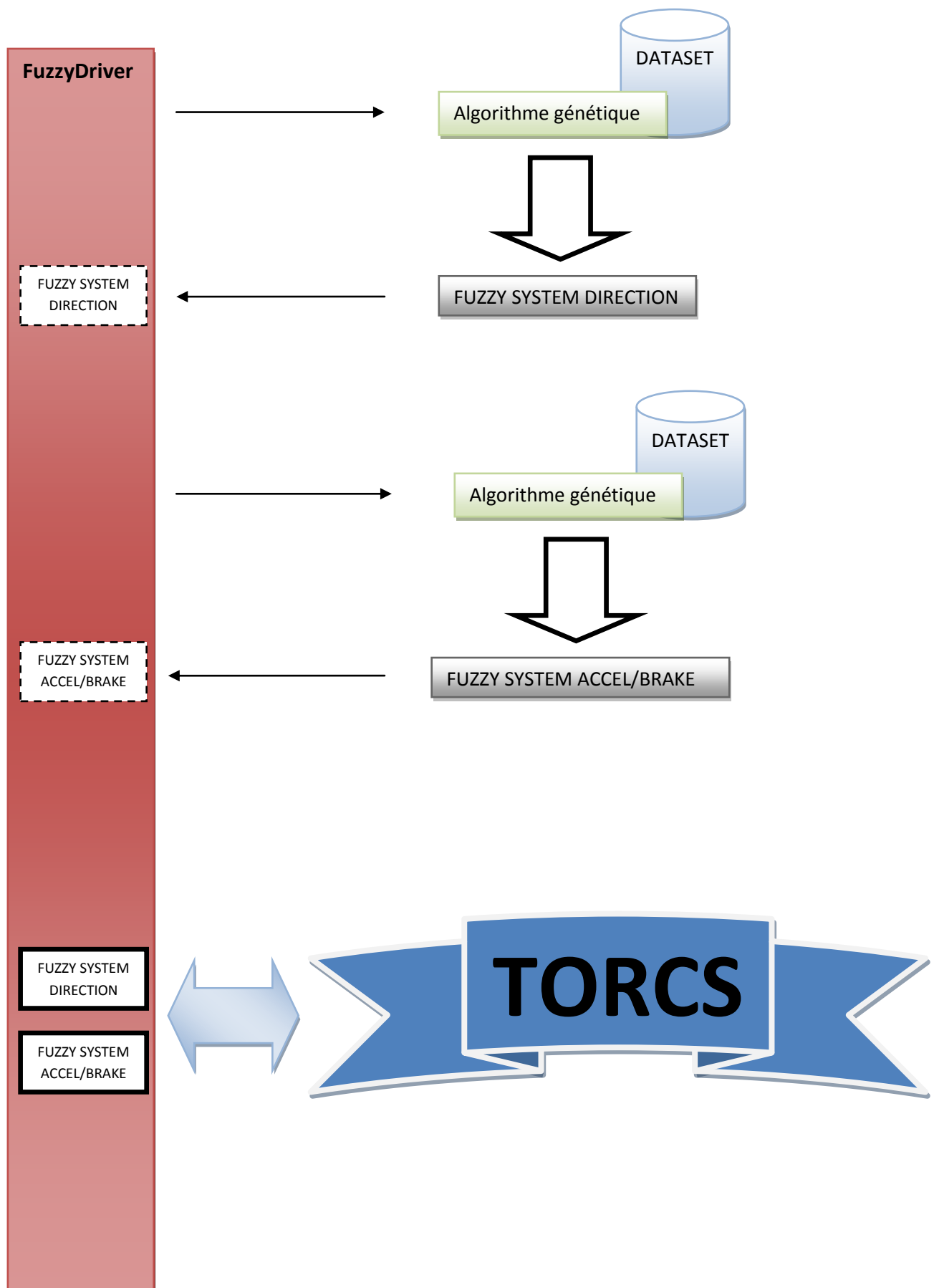
- jFuzzyLogic : Un package permettant la création de systèmes flous et implémentant les spécifications FCL.
- JGAP (Java Genetic Algorithms Package) : JGAP est un package contenant notamment des algorithmes génétiques simples et facilement utilisables.
- Dataset : Données récoltées par 21 capteurs lors du pilotage de la voiture de course par un joueur humain, ainsi que les réactions du joueur (direction et accélération/freinage). (humanPlayerData.csv)

Le travail à réaliser consiste à modifier votre class FuzzyDriver afin de ne plus utiliser vos fichier FCL pour le contrôle de la voiture, mais à générer automatiquement 2 systèmes flous au moyen d'un algorithme génétique.

Conseils:

1. Commencer par définir sur papier le codage de votre chromosome. (variables, règles, fonctions d'appartenance).
2. Implémenter une fonction permettant construire un système flou à partir d'un chromosome. La demo « TestTipperJava.java » du package jFuzzyLogic peut vous être d'une grande aide.
3. Implémenter la fonction de fitness de votre algorithme génétique en utilisant le dataset à disposition. Évaluez votre système flou à l'aide des données des capteurs et comparez les prédictions du système avec les réactions du joueurs.
4. Ajouter un appel à votre algorithme dans le constructeur de FuzzyDriver afin de générer les 2 systèmes flous avant le commencement de la course.

Schéma de fonctionnement :



Dataset “humanPlayerData.csv” specifications:

- The speed of the car divided by 100 (plus some normally distributed noise)
- The orientation of the car w.r.t the circuit in radians (plus some normally distributed noise)
- The distance to the edges of the circuit at 19 different angles w.r.t the car, divided by 100 (plus some normally distributed noise)
- The steering action performed by the user
- The acceleration/braking action performed by the user

Le dataset se trouve à l'adresse :

`\\eint20\Profs\ABJ\LaboEvoTorcs\humanPlayerData.csv`

EvoExample:

Petit programme d'exemple pour l'utilisation de jgap :

`\\eint20\Profs\ABJ\LaboEvoTorcs\EvoExample`

Algorithme génétique essayant de trouver la suite d'entier [0, 1, 2, 3, 4, 5, 6, 7, 8, 9].

Ce petit exemple constitue un bon point de départ pour votre Labo !

Travail à rendre

- Vous devez rendre les fichiers sources permettant de tester votre programme.
- Un PETIT rapport contenant :
 - Une courte introduction expliquant ce qu'est un « algorithme génétique » et un « apprentissage supervisé »
 - Le codage de votre Chromosome avec la position des variables, règles, etc. Ainsi qu'une justification de votre choix de codage
 - Une explication de votre fonction de fitness ainsi qu'une justification
 - Une conclusion

Ces documents doivent être déposés à l'adresse :

`\\eint20\Profs\ABJ\LaboEvoTorcs\retour\[VotreNom]`

Jusqu'au **Mercredi 23 Mai 2012 DERNIER DELAI !**

Barème de correction

Rapport	
Introduction	5%
Codage chromosome	10%
Fonction de fitness	10%
Conclusion	5%
Implémentation	
Représentation d'un système flou sous la forme d'un chromosome.	10%
Reconstruction d'un système floue à partir d'un chromosome.	20%
Fonction de fitness / évaluation d'un système flou	20%
Intégration de l'apprentissage automatique au client Torcs.	10%

